

# tutoriel

June 14, 2022

## 1 introduction to the management of the execution environment with conda

### 1.1 Why using an environment manager?

- avoid compilation and dependencies problems: an environment manager will take care of everything!
- have several environments in parallel, each with their own set of tools (or version)
- useful when cross-tools dependencies are incompatible with each other

### 1.2 Conda definitions

**Environment:** a set of packages/tools in a directory (added to your PATH) **Conda:** an open source package + a general-purpose environment management system (installation, execution, upgrade). For any programming language, multi-platform (Windows, MacOS, Linux) **Conda package:** a compressed tarball of a tool

### 1.3 Conda access

**Conda distribution:** - Anaconda: a data science platform, comes with a lot of packages - Mini-conda3: come without installed packages

Conda is so used that it could even be installed by default to your machine. eg. Conda is: - present in the `jupyter/minimal-notebook` docker container - already activated on the IFB cluster, but to manage some environment variables, activate it (`module load conda`)

Try this in a terminal: `conda --version`

### 1.4 Conda access

**The "conda hub": to download conda package** - Anaconda cloud (private company) relies on the community of developers, concerns many domains (Machine Learning, Data Visualization, Dashboarding-web, Image Processing, Natural Language Processing, etc). On a browser: "conda hub" => <https://anaconda.org> - made up of channels/owners. Each channel contains one or more conda packages - be careful when downloading any packages from an untrusted source, always inspect before installation

### 1.5 About channels

**Some conda channels:** - **default** - **conda-forge:** many popular python packages (analogous to PyPI but with a unified, automated build infrastructure and more peer review of recipes) -

**bioconda:** bioinformaticians' contributions - **private**

**Channels list order** - when different channels have the same package collisions - collisions resolved following the order of your channels list put supplemental channels at the bottom of your channel list

## 1.6 Conda and R

The R interpreter is included in the `r-essentials` packages (200 r-packages). Add `r-` before the regular R package name (eg. `r-ggplot2`)

## 1.7 Mamba

A fast drop-in alternative to conda, using `libsolv` for dependency resolution: just install the mamba package and next, replace all `conda` by `mamba` to use it in conda command

## 1.8 Conda commands

**shell initialisation:** `conda init bash` **creation** of a conda environment: `conda create env -n myenv` **list environments** (\* for the active one): `conda info --envs` **activate** the *myenv* environment: `conda activate myenv` **list packages** (only in an active environment): `conda list` **installation** of a tool/package: `conda install package` **suppress a package** from the environment: `conda remove package` **suppress the myenv environment:** `conda env remove -n myenv` **inactivate** the environment: `conda deactivate`

note: with the `miniconda3` distribution environments are installed by default in a `miniconda3/envs/` repository

## 1.9 2 ways to use conda

**interactive** - create an environment - activate the environment - install some conda packages

**configuration file** - list all conda packages in a configuration file (yml or json format) - create the environment based on the configuration file (option `-f`) - activate the environment

**reproducibility point of view:** use a configuration file and specify a precise version of a package: `<channel>::<package>=<version>`

## 1.10 a practical example

An NGS analysis need the [samtools](#) tool.

These tool is not yet present:

```
[3]: %sh
cd ${PWD}
whereis samtools
```

samtools:

Check the [anaconda web page](#) to find the tool, identify the channel and the version:

and edit a yml file to guide the environment creation:

```
[4]: %sh
cd ${PWD}
echo "name: conda_env_samtools\nchannels:\n - bioconda\ndependencies:\n -
↳bioconda::samtools=1.15.1" > conda_env_samtools.yml
more conda_env_samtools.yml
```

```
:::::::::::
conda_env_samtools.yml
:::::::::::
name: conda_env_samtools
channels:
  - bioconda
dependencies:
  - bioconda::samtools=1.15.1
```

Before creating Conda environment, Conda need to know your shell version. But the `conda init bash` command need a close and re-open the terminal that doesn't work with the undelying terminal opened at the begining of this notebook.

For now, **open a term launcher and copy/paste the notebook command line.**

Before getting these tools and if not already done, initialize your shell for conda (choose bash) and close then reopen a terminal:

```
conda init bash
```

Manage the "envfair" environment: 1) create 2) activate 3) use 4) quit:

```
conda env list
conda env create -f conda_env_samtools.yml #1
conda env list
conda activate conda_env_samtools #2
samtools --help #3
conda deactivate #4
samtools --help
```